



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/657,989	09/09/2003	Shree A. Dandekar	DC-05164	1287
33438	7590	10/03/2006	EXAMINER	
HAMILTON & TERRILE, LLP			CHEN, QING	
P.O. BOX 203518			ART UNIT	PAPER NUMBER
AUSTIN, TX 78720			2191	

DATE MAILED: 10/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/657,989	DANDEKAR ET AL.	
	Examiner	Art Unit	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 09 September 2003.

2a) This action is FINAL. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-20 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-20 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 09 September 2003 is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

- Certified copies of the priority documents have been received.
- Certified copies of the priority documents have been received in Application No. _____.
- Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 20040601.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) Notice of Informal Patent Application

6) Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on September 9, 2003 and the preliminary amendment to the specification filed on May 28, 2004. **Claims 1-20** are currently pending and have been considered below.

Information Disclosure Statement

2. The Office acknowledges receipt of the Information Disclosure Statement filed on June 1, 2004. It has been placed in the application file and the information referred to therein has been considered by the Examiner.

Drawings

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:

- Element 170 in Figure 1; and
- Element 300 in Figure 3.

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application.

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description:

- Element 100 in page 4, line 9; and

- Element 212 in page 5, line 24.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. If so desired, the Applicant is allowed to delete the reference character(s) from the description instead of adding them to the drawing(s).

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The abstract of the disclosure is objected to because the first sentence is missing a verb. It should presumably be read "An automated system for validation, scripting, dissemination and installation of software on information handling systems is disclosed." Correction is required. See MPEP § 608.01(b).

5. The disclosure is objected to because of the following informalities:
 - The specification contains the following typographical errors:
 - The comma (,) in page 2, line 15 should be deleted.
 - For Element 144, “network connection” should be changed to “network connections” in page 5, line 11 in order to be consistent with the descriptive text labels in the drawing.
 - For Element 202, “processor” should be changed to “CPU” in page 5, line 21 in order to be consistent with the descriptive text label in the drawing.
 - For Element 312, “server” should be changed to “repack and script regeneration server” in page 7, line 16 in order to be consistent with the descriptive text label in the drawing.
 - For Element 318, “script and installation validation server” and “server” should be changed to “script and installer validation server” in page 7, lines 20 and 23 in order to be consistent with the descriptive text label in the drawing.
 - The description for step 420 in page 9, lines 2-3 is inconsistent with the descriptive text label in the drawing. It should presumably be read “software package is installed.”

Appropriate correction is required.

Claim Objections

6. **Claims 1, 3, 6-8, 10, 13, 15, 17, and 20** are objected to because of the following informalities:

- **Claims 1, 8, and 15** contain a typographical error: the word “verification” should be changed to “verifying” in the phrase “... said software to a target information handling system after *verification* that said software file complies ...”
- **Claims 1 and 15** recite the limitation “said repack and script server” in lines 5 and 8, respectively. The Examiner subsequently interprets this limitation as reading “said repack and script regeneration server” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 1, 3, 10, 15, and 17** recite the limitation “said software files.” The Examiner subsequently interprets this limitation as reading “said software file” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 3, 6, 7, 17, and 20** contain a typographical error: a comma (,) should be added between the parent claim number and either the word “further” or “wherein” of the limitation clause.
- **Claims 3 and 17** contain a typographical error: “the compliance server” should be changed to “said compliance server.”
- **Claims 6, 13, and 20** contain a typographical error: “the software file” should be changed to “said software file.”
- **Claim 15** contains the following typographical errors:
 - The semicolon (;) at the end of line 5 should be changed to a colon (:).

Art Unit: 2191

- The word “and” should be added after the semicolon (;) in line 2.
- The article “a” should be added in front of the “data storage” limitation in line 3.

Appropriate correction is required.

Double Patenting

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

8. **Claims 1-3, 5-10, 12-17, 19, and 20** are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claims 1, 4-9, 12-17, and 20-24 of copending Application No. 10/768,823 (hereinafter referred to as Application ‘823) in view of Feinman (US 6,075,943).

Claim 1 of Application '823 is compared to Claim 1 of the instant application in the table below. The only additions to the claim in the instant application are a script validation server that generates commands to automatically control the downloading of software images of the software file to a target information handling system and that the software file is transferred to a target information handling system after verification that the software file complies with a set of predetermined parameters.

In the same field of endeavor, Feinman discloses a system and method for remotely transferring and installing client server application programs from a source computer onto a remote client within a data processing system, where a sequential file—located on a server—is used to store information to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point, and the time that each application program is to be delivered (*see Figure 1B, Element 11; Figure 7, Element 100; Column 3, Lines 3-5 and 44-67; and Column 4, Lines 1-4*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate a script validation server that generates commands to automatically control the downloading of software images of the software file to a target information handling system in the system of Application '823, since the system of Application '823 utilizes a download server to transfer the software file to a target information handling system. One would have been motivated to incorporate a script validation server that generates commands to automatically control the downloading of software images of the software file to a target information handling system, since it is not practical for someone to send the application

to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Column 1, Lines 26-32*).

Also, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the software file transferred to a target information handling system after verification that the software file complies with a set of predetermined parameters in the system of Application '823, since the system of Application '823 is operable to transfer the software file to a target information handling system. One would have been motivated to have the software file transferred to a target information handling system after verification that the software file complies with a set of predetermined parameters to ensure that the software meets the compatibility requirements of the computer system and to minimize the occurrence of errors during the subsequent installation of the software file.

Furthermore, Claim 1 of Application '823 recites that the software file comprises a plurality of individual constituent program files and a plurality of installation parameters associated with the software file and that the repack and script regeneration server: a) disassembles the software file into a first plurality of individual constituent program files; b) generates customized installation scripts in accordance with the plurality of parameters associated with the software file; and c) removes predetermined individual constituent program files in accordance with a prune list, thereby, creating a second plurality of individual constituent program files. These limitations are broader than the same limitations recited in Claim 1 of the instant application. Thus, these limitations of Claim 1 of the instant application are anticipated by the same limitations of Claim 1 of Application '823. *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993). Therefore, the recited functional limitations of "a distribution server" and

“a repack and script regeneration server” in Claim 1 of the instant application would be an obvious variation of the recited limitation of “a distribution server” and “a repack and script regeneration server” in Claim 1 of Application ‘823.

Copending Application 10/768,823	Instant Application 10/657,989
1. A system for automated dissemination of software to an information handling system, comprising: a distribution server operable to receive a software file <u>comprising a plurality of individual constituent program files and a plurality of installation parameters associated with said software file;</u>	1. A system for automated dissemination of software to an information handling system, comprising: a distribution server operable to receive a software file;
a repack and script regeneration server operably connected to said distribution server, wherein said repack and script server: a) disassembles said software file <u>into a first plurality of individual constituent program files; b) generates customized installation scripts in accordance with said plurality of parameters associated with said software file; c) removes predetermined individual constituent program files from said plurality of individual constituent program files in accordance with a prune list, thereby creating a second plurality of individual constituent program files;</u> and d) repackages said second plurality of files combined with said customized installation scripts to provide automatic transfer of said software files to an information handling system; and	a repack and script regeneration server operably connected to said distribution server, said repack and script server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software files;
	<u>a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system; and</u>
a download server operable to transfer said second plurality of files to a target information handling system	a download server operable to transfer said software to a target information handling

handling system.	system <u>after verification that said software file complies with a set of predetermined parameters.</u>
------------------	---

Claim 8 of the instant application is an independent method claim that corresponds to Claim 1 of the instant application, and as such is provisionally rejected for the reasons set forth in the provisional double patenting rejection of Claim 1 above.

Copending Application 10/768,823	Instant Application 10/657,989
9. A method for automated dissemination of software to an information handling system, comprising: receiving a software file <u>comprising a plurality of individual constituent program files and a plurality of installation parameters associated with said software file;</u>	8. A method for automated dissemination of software to an information handling system, comprising: receiving a software file;
disassembling said software file <u>into a plurality of individual constituent program files;</u>	disassembling said software file and
<u>generating customized installation scripts in accordance with said parameters associate with said software file;</u>	
<u>removing predetermined individual constituent program files from said plurality of individual constituent program files in accordance with a prune list, thereby creating a second plurality of individual constituent program files;</u>	
repacking said second plurality of files combined with said customized installation scripts to provide automatic transfer of said software files to an information handling system; and	repackaging said software file with scripts for automatically controlling the transfer of said software file;
	<u>generating commands to control the automatic downloading of software images of said software file to a target information handling system; and</u>
transferring said second plurality of files to a target information handling system.	transferring said software to a target information handling system <u>after verification</u>

	<u>that said software file complies with a set of predetermined parameters.</u>
--	---

Claim 15 of the instant application is an independent system claim that corresponds to Claim 1 of the instant application, and as such is provisionally rejected for the reasons set forth in the provisional double patenting rejection of Claim 1 above.

Copending Application 10/768,823	Instant Application 10/657,989
17. An information handling system, comprising: a data processor;	15. An information handling system, comprising: a data processor;
data storage having a software file stored thereon, wherein said software is transferred to said data storage by an automated software dissemination system comprising: a distribution server operable to receive a software file <u>comprising a plurality of individual constituent program files and a plurality of installation parameters associated with said software file;</u>	data storage having a software file stored thereon, said software file being transferred to said data storage by an automated software dissemination system comprising: a distribution server operable to receive a software file;
a repack and script regeneration server operably connected to said distribution server, wherein said repack and script server: a) disassembles said software file <u>into a first plurality of individual constituent program files; b) generates customized installation scripts in accordance with said plurality of parameters associated with said software file; c) removes predetermined individual constituent program files from said plurality of individual constituent program files in accordance with a prune list, thereby creating a second plurality of individual constituent program files;</u> and d) repackages said second plurality of files combined with said customized installation scripts to provide automatic transfer of said software files to an information handling system; and	a repack and script regeneration server operably connected to said distribution server, said repack and script server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software files;
	<u>a script validation server operably coupled to said repack and script regeneration</u>

	<u>server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system; and</u>
a download server operable to transfer said second plurality of files to a target information handling system.	a download server operable to transfer said software to said information handling system <u>after verification that said software file complies with a set of predetermined parameters.</u>

As per Claims 2, 9, and 16 of the instant application, these claims very broadly recite a compliance server operable to perform compliance verification to confirm that the software file complies with the set of predetermined parameters, which would have been obvious to one of ordinary skill in the art at the time the invention was made, since it is further suggested in Claims 5, 13, and 21 of Application '823. One would have been motivated to incorporate a script validation server that generates commands to perform compliance verification to ensure that the software meets the compatibility requirements of the computer system and to minimize the occurrence of errors during the subsequent installation of the software file.

As per Claims 3, 5-7, 10, 12-14, 17, 19, and 20 of the instant application, the limitations in each of these dependent claims are also recited in Claims 5-8, 13-16, and 21-24, respectively, of Application '823.

This is a provisional obviousness-type double patenting rejection.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. **Claims 1-20** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 8, and 15 recite the limitation “said software.” There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “said software file” for the purpose of further examination.

Claims 2-7, 9-14, and 16-20 depend on Claims 1, 8, and 15, respectively. Therefore, these claims suffer the same deficiency as their respective parent claims.

Claims 6, 13, and 20 recite the limitation “the software distribution system.” There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “a software distribution system” for the purpose of further examination.

Claims 7 and 14 recite the limitation “said repackaged software file” in lines 3 and 2, respectively. There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “said software file” for the purpose of further examination.

Claim Rejections - 35 USC § 101

11. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

12. **Claim 12** is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The result of **Claim 12** is directed to the act of “verifying,” which does not appear to be a tangible result so as to constitute a practical application of the idea. The act of “verifying” is merely a thought or an abstract idea and does not appear to produce a tangible result even if the step of verification does occur, since the result of that verification is not conveyed in the real world. The result is a verification, which is neither used in a disclosed practical application nor made available for use in a disclosed practical application. It also does not appear that the usefulness of the verification can be realized from the claimed steps to support a disclosed specific, substantial, and credible utility so as to produce a useful result.

Therefore, the claim does not meet the statutory requirement of 35 U.S.C. § 101, since the claim is not directed to a practical application of the § 101 judicial exception producing a result tied to the physical world.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

14. **Claims 1-3, 5, 6, 8-10, 12, 13, 15-17, 19, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over Amberg et al. (US 5,991,543) in view of Feinman (US 6,075,943).

As per **Claim 1**, Amberg et al. disclose a system for automated dissemination of software to an information handling system, comprising:

- A. A distribution server operable to receive a software file (*see Figure 1, Element 140; and Column 3, Lines 48-55 and 59-66*);
- B. A repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1, Element 140; Column 4, Lines 8-14; Column 9, Lines 56-67; and Column 10, Lines 1-2*); and

C. A download server operable to transfer said software file to a target information handling system after verifying that said software file complies with a set of predetermined parameters (*see Figure 1, Elements 170 and 190; Figure 2, Element 202, and Column 10, Lines 61-64*).

However, Amberg et al. do not explicitly disclose that the system comprising a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system.

In the same field of endeavor, Feinman discloses a system and method for remotely transferring and installing client server application programs from a source computer onto a remote client within a data processing system, where a sequential file—located on a server—is used to store information to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point, and the time that each application program is to be delivered (*see Figure 1B, Element 11; Figure 7, Element 100; Column 3, Lines 3-5 and 44-67; and Column 4, Lines 1-4*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to have a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system in the system of Amberg et al., since Amberg et al. already disclose

that the network connection of the system can be a local area network, an intranet, or the Internet and that the output files are downloaded to the target computer system (*see Column 4, Lines 3-7; and Column 10, Lines 61-64*). One would have been motivated to incorporate a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Column 1, Lines 26-32*).

As per **Claim 2**, Amberg et al., as modified by Feinman, disclose a system for automated dissemination of software to an information handling system **as in Claim 1 above**, and Amberg et al. further disclose that the system comprising a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with said set of predetermined parameters (*see Figure 1, Element 100; and Column 9, Lines 9-16*).

As per **Claim 3**, Amberg et al., as modified by Feinman, disclose a system for automated dissemination of software to an information handling system **as in Claim 2 above**, and Amberg et al. further disclose that said compliance server is operable to automatically generate a non-compliance notice message upon detection that said software file does not comply with said predetermined set of parameters (*see Figure 1, Element 100; and Column 9, Lines 9-16*).

As per **Claim 5**, Amberg et al., as modified by Feinman, disclose a system for automated dissemination of software to an information handling system **as in Claim 1 above**, and further disclose that the system comprising a test control server operable to verify proper operation of said software file on said target information handling system (*see Figure 11; Column 12, Lines 1-3 and 62-67; and Column 13, Lines 1-12*), but Amberg et al. do not explicitly disclose that the system comprising a test control server operable to confirm the download of said software file to said target information handling system. However, Feinman does disclose a test control server operable to confirm the download of said software file to said target information handling system (*see Figure 5, Element 70; Column 3, Lines 40-43; and Column 5, Lines 18-21*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to allow the test control server operable to confirm the download of said software file to said target information handling system in the system of Amberg et al., since Amberg et al. already disclose that the output files are downloaded to the target computer system (*see Column 4, Lines 3-7; and Column 10, Lines 61-64*). One would have been motivated to allow the test control server operable to confirm the download of said software file to said target information handling system so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Column 1, Lines 33-37*).

As per **Claim 6**, Amberg et al., as modified by Feinman, disclose a system for automated dissemination of software to an information handling system **as in Claim 1 above**, and Amberg

et al. further disclose that said distribution server is operable to notify a manager regarding the status of said software file within a software distribution system (*see Column 14, Lines 20-25*).

As per **Claim 8**, Amberg et al. disclose a method for automated dissemination of software to an information handling system, comprising:

- A. Receiving a software file (*see Figure 1, Element 140; and Column 3, Lines 48-55 and 59-66*);
- B. Disassembling said software file and repackaging said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1, Element 140; Column 4, Lines 8-14; Column 9, Lines 56-67; and Column 10, Lines 1-2*); and
- C. Transferring said software file to a target information handling system after verifying that said software file complies with a set of predetermined parameters (*see Figure 1, Elements 170 and 190; Figure 2, Element 202, and Column 10, Lines 61-64*).

However, Amberg et al. do not explicitly disclose that the method comprising the step of generating commands to control the automatic downloading of software images of said software file to a target information handling system.

In the same field of endeavor, Feinman discloses a system and method for remotely transferring and installing client server application programs from a source computer onto a remote client within a data processing system, where a sequential file—located on a server—is used to store information to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point, and the time that each application program is to be

delivered (*see Figure 1B, Element 11; Figure 7, Element 100; Column 3, Lines 3-5 and 44-67; and Column 4, Lines 1-4*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to generate commands to control the automatic downloading of software images of said software file to a target information handling system in the system of Amberg et al., since Amberg et al. already disclose that the network connection of the system can be a local area network, an intranet, or the Internet and that the output files are downloaded to the target computer system (*see Column 4, Lines 3-7; and Column 10, Lines 61-64*). One would have been motivated to generate commands to control the automatic downloading of software images of said software file to a target information handling system, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Column 1, Lines 26-32*).

As per **Claim 9**, Amberg et al., as modified by Feinman, disclose a method for automated dissemination of software to an information handling system as in **Claim 8 above**, and Amberg et al. further disclose that the method comprising the step of performing compliance verification to confirm that said software file complies with said set of predetermined parameters (*see Figure 1, Element 100; and Column 9, Lines 9-16*).

As per **Claim 10**, Amberg et al., as modified by Feinman, disclose a method for automated dissemination of software to an information handling system as in **Claim 9 above**,

and Amberg et al. further disclose that the method comprising the step of generating a non-compliance notice message upon detection that said software file does not comply with said predetermined set of parameters (*see Figure 1, Element 100; and Column 9, Lines 9-16*).

As per **Claim 12**, Amberg et al., as modified by Feinman, disclose a method for automated dissemination of software to an information handling system **as in Claim 8 above**, and further disclose that the method comprising the step of verifying proper operation of said software file on said target information handling system (*see Figure 11; Column 12, Lines 1-3 and 62-67; and Column 13, Lines 1-12*), but Amberg et al. do not explicitly disclose that the method comprising the step of confirming the download of said software file to said target information handling system. However, Feinman does disclose the step of confirming the download of said software file to said target information handling system (*see Figure 5, Element 70; Column 3, Lines 40-43; and Column 5, Lines 18-21*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to confirm the download of said software file to said target information handling system in the system of Amberg et al., since Amberg et al. already disclose that the output files are downloaded to the target computer system (*see Column 4, Lines 3-7; and Column 10, Lines 61-64*). One would have been motivated to confirm the download of said software file to said target information handling system so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Column 1, Lines 33-37*).

As per **Claim 13**, Amberg et al., as modified by Feinman, disclose a method for automated dissemination of software to an information handling system as in **Claim 8 above**, and Amberg et al. further disclose that the method comprising the step of notifying a manager regarding the status of said software file within a software distribution system (*see Column 14, Lines 20-25*).

As per **Claim 15**, Amberg et al. disclose an information handling system, comprising:

- A. A data processor (*see Column 3, Lines 32-37*); and
- B. A data storage having a software file stored thereon (*see Column 3, Lines 32-37*), said software file being transferred to said data storage by an automated software dissemination system comprising:
 - I. A distribution server operable to receive a software file (*see Figure 1, Element 140; and Column 3, Lines 48-55 and 59-66*);
 - II. A repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1, Element 140; Column 4, Lines 8-14; Column 9, Lines 56-67; and Column 10, Lines 1-2*); and
 - III. A download server operable to transfer said software file to said information handling system after verifying that said software file complies with a set of

predetermined parameters (*see Figure 1, Elements 170 and 190; Figure 2, Element 202, and Column 10, Lines 61-64*).

However, Amberg et al. do not explicitly disclose that the automated software dissemination system comprising a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system.

In the same field of endeavor, Feinman discloses a system and method for remotely transferring and installing client server application programs from a source computer onto a remote client within a data processing system, where a sequential file—located on a server—is used to store information to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point, and the time that each application program is to be delivered (*see Figure 1B, Element 11; Figure 7, Element 100; Column 3, Lines 3-5 and 44-67; and Column 4, Lines 1-4*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to have a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system in the system of Amberg et al., since Amberg et al. already disclose that the network connection of the system can be a local area network, an intranet, or the Internet and that the output files are downloaded to the target computer system (*see Column 4, Lines 3-7;*

and Column 10, Lines 61-64). One would have been motivated to incorporate a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Column 1, Lines 26-32*).

As per **Claim 16**, Amberg et al., as modified by Feinman, disclose an information handling system **as in Claim 15 above**, and Amberg et al. further disclose that the system comprising a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with said set of predetermined parameters (*see Figure 1, Element 100; and Column 9, Lines 9-16*).

As per **Claim 17**, Amberg et al., as modified by Feinman, disclose an information handling system **as in Claim 16 above**, and Amberg et al. further disclose that said compliance server is operable to automatically generate a non-compliance notice message upon detection that said software file does not comply with said predetermined set of parameters (*see Figure 1, Element 100; and Column 9, Lines 9-16*).

As per **Claim 19**, Amberg et al., as modified by Feinman, disclose an information handling system **as in Claim 15 above**, and further disclose that the system comprising a test control server operable to verify proper operation of said software file on said target information handling system (*see Figure 11; Column 12, Lines 1-3 and 62-67; and Column 13, Lines 1-12*), but Amberg et al. do not explicitly disclose that the system comprising a test control server operable to confirm the download of said software file to said target information handling system. However, Feinman does disclose a test control server operable to confirm the download of said software file to said target information handling system (*see Figure 5, Element 70; Column 3, Lines 40-43; and Column 5, Lines 18-21*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to allow the test control server operable to confirm the download of said software file to said target information handling system in the system of Amberg et al., since Amberg et al. already disclose that the output files are downloaded to the target computer system (*see Column 4, Lines 3-7; and Column 10, Lines 61-64*). One would have been motivated to allow the test control server operable to confirm the download of said software file to said target information handling system so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Column 1, Lines 33-37*).

As per **Claim 20**, Amberg et al., as modified by Feinman, disclose an information handling system **as in Claim 15 above**, and Amberg et al. further disclose that said distribution

server is operable to notify a manager regarding the status of said software file within a software distribution system (*see Column 14, Lines 20-25*).

15. **Claims 4, 11, and 18** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Amberg et al. (US 5,991,543)** in view of **Feinman (US 6,075,943)** as applied to Claims 1, 8, and 15 above, and further in view of **Tso et al. (US 6,088,803)**.

As per **Claim 4**, Amberg et al., as modified by Feinman, disclose a system for automated dissemination of software to an information handling system **as in Claim 1 above**, but neither reference discloses that said distribution server is operable to scan said software file for viruses.

However, in the same area the problem sought to be solved, Tso et al. disclose a system for virus checking a data object to be downloaded to a client service, where the network device retrieves the requested data object from client server and invokes virus checker to perform its preconfigured virus scan processing with the requested file as input (*see Figure 2, Element 40; Column 2, Lines 62-67; and Column 3, Lines 1-5*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to allow said distribution server to scan said software file for viruses in the system of Amberg et al., since it is a common practice to scan a software file for viruses prior to and/or after downloading it. One would have been motivated to allow said distribution server to scan said software file for viruses in order to minimize breaches in system integrity (*see Column 1, Lines 27-28*).

As per **Claim 11**, Amberg et al., as modified by Feinman, disclose a method for automated dissemination of software to an information handling system **as in Claim 8 above**, but neither reference discloses that the method comprising the step of scanning said software file for viruses.

However, in the same area the problem sought to be solved, Tso et al. disclose a system for virus checking a data object to be downloaded to a client service, where the network device retrieves the requested data object from client server and invokes virus checker to perform its preconfigured virus scan processing with the requested file as input (*see Figure 2, Element 40; Column 2, Lines 62-67; and Column 3, Lines 1-5*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to scan said software file for viruses in the system of Amberg et al., since it is a common practice to scan a software file for viruses prior to or after downloading it. One would have been motivated to scan said software file for viruses in order to minimize breaches in system integrity (*see Column 1, Lines 27-28*).

As per **Claim 18**, Amberg et al., as modified by Feinman, disclose an information handling system **as in Claim 15 above**, but neither reference discloses that said distribution server is operable to scan said software file for viruses.

However, in the same area the problem sought to be solved, Tso et al. disclose a system for virus checking a data object to be downloaded to a client service, where the network device retrieves the requested data object from client server and invokes virus checker to perform its

preconfigured virus scan processing with the requested file as input (*see Figure 2, Element 40; Column 2, Lines 62-67; and Column 3, Lines 1-5*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to allow said distribution server to scan said software file for viruses in the system of Amberg et al., since it is a common practice to scan a software file for viruses prior to or after downloading it. One would have been motivated to allow said distribution server to scan said software file for viruses in order to minimize breaches in system integrity (*see Column 1, Lines 27-28*).

16. **Claims 7 and 14** are rejected under 35 U.S.C. 103(a) as being unpatentable over Amberg et al. (US 5,991,543) in view of Feinman (US 6,075,943) as applied to Claims 1 and 8 above, and further in view of Karasudani et al. (US 6,378,054).

As per **Claim 7**, Amberg et al., as modified by Feinman, disclose a system for automated dissemination of software to an information handling system **as in Claim 1 above**, but neither reference disclose that the system comprising an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon.

In the same area the problem sought to be solved, Karasudani et al. disclose a technique related to data backup for a computer, where data files are added to an archive database (*see Column 11, Lines 60-61*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Karasudani et al. into the teaching of Amberg et al. to have an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon in the system of Amberg et al., since the system of Amberg et al. already utilizes a database to store software installation files and/or testing steps (*see Figure 1, Element 100; and Column 4, Lines 1-3*). One would have been motivated to incorporate an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon in order to minimize damages by immediately recovering a data file in the event of problems, such as the loss of data (*see Column 1, Lines 26-29*).

As per **Claim 14**, Amberg et al., as modified by Feinman, disclose a method for automated dissemination of software to an information handling system as in **Claim 8 above**, but neither reference disclose that the method comprising the step of transferring copies of said software file to an archive server for storage thereon.

In the same area the problem sought to be solved, Karasudani et al. disclose a technique related to data backup for a computer, where data files are added to an archive database (*see Column 11, Lines 60-61*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Karasudani et al. into the teaching of Amberg et al. to transfer copies of said software file to said archive server for storage thereon in the system of Amberg et al., since the system of Amberg et al. already utilizes a database to store

software installation files and/or testing steps (*see Figure 1, Element 100; and Column 4, Lines 1-3*). One would have been motivated to transfer copies of said software file to said archive server for storage thereon in order to minimize damages by immediately recovering a data file in the event of problems, such as the loss of data (*see Column 1, Lines 26-29*).

Conclusion

17. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- A. Kroening et al. (US 6,080,207) disclose a system and method of creating and delivering software to the computer systems using an image builder.
- B. Henry (US 6,131,192) discloses an executable file that contains the software materials necessary for the installation of a software product.
- C. Doran, Jr. et al. (US 6,385,766) disclose a method and apparatus for Windows® based installation of software in a build-to-order computer system.
- D. Vrhel, Jr. et al. (US 6,543,047) disclose a method and apparatus for testing custom-configured software/hardware integration in a computer system build-to-order manufacturing process.
- E. Valys (US 6,549,914) discloses a system and method for statistical file preload for factory-installed software.
- F. Barajas et al. (US 6,550,062) disclose a method for installing and testing software for a computer system.

G. Bryan et al. (US 6,591,418) disclose a software installation system is provided which allows only the most recently used software to be stored on the file servers while the entire set of available software is resident on a library server.

H. Vrhel, Jr. et al. (US 6,598,223) disclose a system and method for installing and testing build-to-order components as part of an image-based software installation in a computer system.

I. Bearden et al. (US 6,718,373) disclose a system and method for installing files in a computing system.

J. Fogarty et al. (US 6,721,946) disclose a system and method for installing software on a computer.

K. Cohen et al. (US 6,938,250) disclose a system and method for image-based software installation.

L. Lin (US 2002/0087965) discloses a computer manufacturer and a computer company separately delivers assembled hardware components and a pre-loaded storage device to a client, and the client installs the pre-loaded storage device in the assembled hardware components to complete a computer.

M. Ream et al. (US 2002/0112232) disclose an automated installation of software onto recipient computers to allow the recipient computers to be prepared in a rapid and efficient manner.

N. Lu et al. (US 2002/0156877) disclose a system and method for duplicating a software system and installing that software system on a target computer.

O. Bryan et al. (US 2002/0188942) disclose a system for factory installation of software that configures the storage resources in one or more file servers as a “cache.”

P. Amro et al. (US 2003/0028629) disclose a system and method for installing software onto a build-to-order computer system using an identification device.

Q. Lin et al. (US 2003/0192043) disclose a method for installing a software bundle on a computer that facilitates installation of software bundles on a plurality of target computers.

R. Sedlack et al. (US 2004/0025155) disclose a system, method, and computer program product for configuring a software image for installation into a computer system.

S. Barajas et al. (US 2005/0055688) disclose a method for automatically installing a software image onto an information handling system.

T. Dandekar et al. (US 2005/0172284) disclose a system and method for automatically generating customized factory installable software.

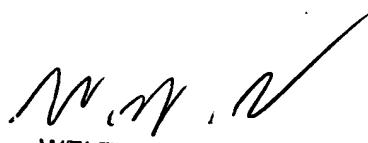
Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications

Art Unit: 2191

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



WEI ZHEN
SUPERVISORY PATENT EXAMINER

QC / QC
September 25, 2006